Klasifikasi Perubahan Perangkat Lunak pada *Mobile App Review* dengan Menggunakan Metode Long Short Term Memory (LSTM)

Alifia Puspaningrum[1], Munengsih Sari Bunga[2], Iryanto[3]

[1,2,3]Politeknik Negeri Indramayu
Jl. Lohbener Lama No.08, Lohbener, Indramayu
E-mail : [1]*alifia.puspaningrum@gmail.com*, [2]*nengslim85@gmail.com*, [3]*iryanto@polindra.ac.id*

ABSTRAK

Seiring dengan perkembangan aplikasi bergerak, perbaikan dan evolusi perangkat lunak menjadi salah satu hal yang wajib untuk dilakukan. Salah satu input yang dapat digunakan dalam proses tersebut diantaranya adalah pengalaman pengguna dalam menggunakan produk. Beberapa jenis kategori perubahan perangkat lunak yang sering digunakan sebagai pemetaan diantaranya adalah *Bug Error, Feature Request*, dan *Non Informative.* Penelitian sebelumnya mengelompokan perubahan perangkat lunak dengan menganalisa nilai similaritas dari hidden topics yang dihasilkan oleh Latent Dirichlet Allocation (LDA). Namun performa dari pelabelan tidak terlalu baik karena hanya mempertimbangkan nilai similaritas beberapa term yang mewakili kalimat ulasan saja. Oleh karena itu, penelitian ini mengusulkan metode yang mempertimbangkan *similarity clustering* dan *lexical analysis* dari dokumen ulasan yang selanjutnya diklasifikasi dengan menggunakan Long Short Term Memory (LSTM) pada kategori Bug Report dengan nilai 93.1% untuk akurasi, 100% untuk precision, dan 93.1% untuk recall.

Kata kunci : ***Aplikasi Bergerak, Labeling Teks, Perbaikan Perangkat Lunak, Skor Global, Ulasan***

ABSTRACT

Along with the development of mobile applications, software evolution is an essential step to be done. Mining user experience about the product is one of strategies to obtain many information about the features. Software change categories that are often used are Bug Error, Feature Request, and Non Informative. Previous research categorized software changes by analyzing the similarity of hidden topics produced by Latent Dirichlet Allocation (LDA). But the performance of labeling is not good enough because it only considers the similarity value of some terms that represent the review sentence. Therefore, this study proposes a method that considers similarity clustering value and lexical analysis of the document, and classify using Long Short Term Memory (LSTM) then. The experimental result shows the best classification for Bug Report software change categories by reaching 93.1% for accuracy, 100% for precision, and 93.1% for recall.

Keyword : Lembaga Mobile Application, Text Labeling, Software Maintenance, Global Score, Review

1. PENDAHULUAN

Saat ini, perkembangan aplikasi mobile telah berkembang pesat (Chen, Lin, Hoi, Xiao, & Zhang, 2014; Hu & Liu, 2004). Meningkatnya pengguna ponsel yang mengunduh aplikasi selama beberapa tahun menunjukkan hal ini fenomena. Kondisi ini mendorong para developer untuk rutin melakukan maintenance software yang mereka kembangkan. Beberapa pengguna memilih untuk menghapus aplikasi mereka karena berbagai faktor, mis. antarmuka pengguna yang buruk, pemberitahuan yang mengganggu, dan proses pendaftaran yang rumit. Oleh karena itu, harus ada metode analisis perangkat lunak yang dapat memberikan fitur rekomendasi yang perlu dikembangkan atau diperbaiki pada siklus pengembangan selanjutnya (Carre & Winbladh, 2013).

Analisa pengalaman pengguna dalam menggunakan produk perlu untuk dianalisa lebih lanjut, sehingga informasi yang diberikan oleh pengguna dalam bentuk review produk dapat diolah untuk dijadikan masukan oleh pengembang aplikasi (Y. Liu, Jin, Ji, Harding, & Fung, 2013). Review produk juga dapat membantu desainer untuk memahami kebutuhan serta preferensi pengguna dalam memutuskan untuk membeli atau tidak (Y. Liu, Lu, & Loh, 2007). Sejak tahun 2003, peneliti mulai menganalisa dan mengembangkan model inovatif dalam mengembangkan analisa review produk yang saat ini dikenal sebagai penggalian review (Hu, Liu, & Street, 2004), (B. Liu, Street, Street, & Street, n.d.), (Xianghua, Guo, Yanyan, & Zhiqiang, 2013).

Review produk yang diberikan oleh pengguna tentunya membutuhkan suatu metode untuk memetakan input sehingga dapat menjadi sistem pendukung dalam mengambil keputusan. (Maalej, 2015) mengklasifikasikan dokumen *review* ke dalam beberapa kategori, seperti *bug reports, feature request, user experience,* dan rating untuk mengekspresikan pengalaman pengguna dalam menggunakan aplikasi.

Beberapa metode telah berhasil dikembangkan untuk mengklasifikasi suatu review ke dalam salah satu jenis dari kategori perubahan perangkat lunak baik dengan menggunakan metode *supervised* atau pun *unsupervised*. Metode *supervised* seperti Support Vector Machine (Samad, Basari, Hussin, Pramudya, & Zeniarja, 2013), Logistic Regression (Hamdan, Bellot, & Bechet, 2015) yang diterapkan pada penggalian review produk membutuhkan data dan pelabelan dalam jumlah besar yang menyebabkan penemuan *ground truth* dari dokumen yang dianalisa tidak dapat dilakukan secara manual. Selain itu, metode ini masih menggunakan fitur statistik yang berbasis pada kemunculan term sebagai fitur pada proses klasifikasi sehingga tidak mempertimbangkan aspek semantik dari dokumen. Adapun metode *unsupervised* pun menunjukan hasil yang tidak terlalu signifikan (Zhai, Liu, Xu, & Jia, 2011). Salah satu metode *unsupervised* yang kerap digunakan dalam penggalian review online adalah topic modelling. Latent Dirichlet Allocation (LDA) menjadi salah satu metode yang paling banyak dikembangkan dalam membentuk model dari topik yang ada (Brody & Elhadad, 2010).

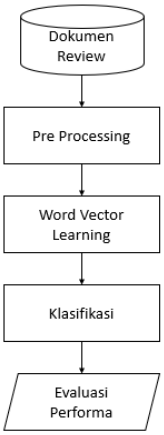
Pada beberapa tahun terakhir, Deep Learning mampu menarik perhatian pada 20 sektor dari pengolahan bahasa manusia, yang secara umum terbagi ke dalam dua area besar (Wang et al., 2016). Sektor pertama adalah mempelajari word embedding dengan melatikan training pada model bahasa (Bengio, Ducharme, Vincent, & Jauvin, 2003), (Mikolov, Corrado, Chen, & Dean, 2013), (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), sedangkan sektor kedua adalah melakukan komposisi semantik untuk memperoleh tingkat representasi dari frasa atau kalimat (Collobert et al., 2011). Salah satu pendekatan deep learning yang mampu secara otomatis mempelajari fitur yang dideskripsikan dalam bentuk vektor adalah Recurrent Neural Networks (RNN). RNN dengan menggunakan word embedding mampu sukses diaplikasikan dalam kasus penggalian review tanpa memerlukan modifikasi proses di dalamnya. Namun, RNN memiliki kelemahan dalam memahami keterhubungan dari suatu sequence yang terpisah dalam jarak yang cukup jauh. Long short term memory (LSTM) pada RNN didesain untuk mampu menutupi kekurangan RNN dalam memodelkan dependesi term yang cukup jauh (Hochreiter & Schmidhuber,

1997). Hal ini membuat metode ini sukses diterapkan pada berbagai sektor, diantaranya pemodelan bahasa, pengenalan suara, dan pemahaman bahasa yang diucapkan.

Penelitian ini bertujuan untuk mengembangkan klasifikasi dalam penggalian review pengguna aplikasi bergerak untuk perbaikan serta evolusi perangkat lunak dengan menggunakan metode LSTM.

2. METODE PENELITIAN

Seperti yang ditunjukan pada Gambar 1, metode penelitian yang dilaksanakan adalah sebagai berikut:



Gambar 1. Bagan Alur Tahapan Penelitian

Pada penelitian yang ditunjukan pada gambar di atas, dokumen review pertama-tama akan dibersihkan pada proses *pre processing*. Setelah bersih, untuk memperoleh fitur vektor, teks akan dirubah menjadi vektor. Selanjutnya vektor tersebut yang akan dipecah menjadi data training dan data testing untuk diklasifikasi dengan LSTM. Performa klasifikasi selanjutnya akan dievaluasi dengan menggunakan akurasi.

3. LANDASAN TEORI

3.1 Pengolahan Bahasa Manusia

Pemrosesan bahasa alami merupakan teknik untuk mengajarkan komputer dalam memahami maksud dari kata-kata yang digunakan oleh manusia. Metode ini lah yang kemudian diadaptasi oleh bidang Rekaya Perangkat Lunak salah satu nya dalam memproses kategori perubahan dari suatu produk perangkat lunak.

Salah satu teknik pemrosesan bahasa alami yang kerap digunakan pada pra proses adalah teknik-teknik berikut, yaitu:

- Tokenisasi
  Pada tahap ini, input teks dokumen dipecah menjadi unit atomis terkecil. Biasanya unit tersebut berupa kata-kata atau kalimat atau paragraf.
- Normalisasi
  Merubah semua huruf menjadi huruf kecil
- Penghilangan Tanda Baca
  Menghilangkan tanda baca pada kalimat
- *Stemming*
  Stemming memiliki peran untuk menjadikan teks menjadi kata dasar.
- *Stopwords Removal*
  Stopwords Removal memiliki peran untuk menghapus kata henti dalam bahasa inggris.
- *Spelling Correction*
  Proses ini memiliki fungsi untuk menyempurnakan kalimat yang memiliki kesalahan dalam penulisan.

3.2 Word Vector Learning

Word embedding merupakan kumpulan nama dari pemodelan bahasa dan teknik ekstraksi fitur pada natural language processing (NLP) dimana setiap kata atau phrasa dari suatu kosakata akan dipetakan menjadi vektor yang berupa bilangan real. Word embedding kerap digunakan dalam neural networks, reduksi dimensi pada matriks kemunculan kata, model probabilistik, dll. Metode word embedding ini juga digunakan sebagai input untuk meningkatkan performa pada pengolahan bahasa manusia seperti parsing sintaktik dan analisa sentimen.

3.3 Long Short Term Memory

LSTM (Hochreiter & Schmidhuber, 1997). adalah arsitektur recurrent neural network (RNN) yang didesain untuk memodelkan keterhubungan antara term yang memiliki interval yang jauh. LSTM telah digunakan

secara luas dalam pengolahan bahasa manusia seperti pada analisa sentimen, parsing sintaksis, kategorisasi dokumen yang memiliki ukuran yang panjang, dll.

3.4 Akurasi

Terdapat banyak macam-macam metode evaluasi yang dapat digunakan untuk mengukur kehandalan dari metode yang diusulkan, salah satunya akurasi. Adapun proses perhitungan dari akurasi, precision, dan recall ditentukan dari prediksi informasi terhadap nilai sebenarnya yang direpresentasikan dengan True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) yang ditunjukan pada Persamaan 1, 2, dan 3.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

4. HASIL DAN PEMBAHASAN

Berdasarkan uji coba yang dilakukan, metode klasifikasi LSTM dibandingkan dengan metode Deep Learning lainnya yaitu Convolutional Neural Network (CNN) dengan membandingkan dengan tiga skenario banyak data, yaitu 300 data, 544 data, dan 884 data. Berdasarkan skenario tersebut, kedua metode klasifikasi tersebut dilihat performa nya untuk masing-masing kategori perubahan perangkat lunak, yaitu Bug Report, Feature Request, dan Non Informative.

Tabel 1. Performa klasifikasi dengan metode LSTM

		LSTM		
		300 Data	544 Data	884 Data
Bug Report	Accuracy	85,37	84,56	**93,10**
	Precision	100,00	100,00	**100,00**
	Recall	85,37	84,56	**93,10**
Feature Request	Accuracy	**98,78**	97,79	94,83
	Precision	**50,00**	0,00	0,00
	Recall	**100,00**	0,00	0,00
Non Informative	Accuracy	87,80	86,76	69,66
	Precision	0,00	0,00	0,00
	Recall	0,00	0,00	0,00

Tabel 2. Performa klasifikasi dengan metode CNN

		CNN		
		300 Data	544 Data	884 Data
Bug Report	Accuracy	84,15	88,24	82,18
	Precision	97,14	97,39	84,57
	Recall	86,08	89,60	95,80
Feature Request	Accuracy	97,56	97,79	91,95
	Precision	0,00	0,00	11,11
	Recall	0,00	0,00	14,29
Non Informative	Accuracy	86,59	**89,71**	75,86
	Precision	10,00	**50,00**	48,00
	Recall	33,33	**64,29**	60,00

Berdasarkan Tabel 1 dan 2 Performa metode secara umum menunjukan bahwa LSTM mampu lebih unggul dibandingkan dengan CNN dalam mengklasifikasi dokumen baik yang menggunakan GloVe maupun tidak.

Jika melihat performa metode untuk setiap kelas, dapat dilihat bahwa pada kelas Bug Report, 884 Data menjadi data yang memperoleh nilai akurasi klasifikasi terbaik yaitu 93,10%. Kelas ini menunjukan bahwa, semakin bertambahnya data, tidak membuat nilai akurasi dari klasifikasi berkurang. Hal ini menunjukan bahwa meskipun data yang digunakan pada data 884 merupakan data yang multi label, LSTM mampu dengan optimal mengklasifikasi. Namun, penambahan data dari 300 data menjadi 544 data menunjukan penurunan performa dari kedua metode klasifikasi, meskipun LSTM mampu menunjukan performa yang lebih baik dibandingkan dengan CNN. Selain itu, perolehan skor precision dan recall pada kelas ini memperlihatkan bahwa nilai true positive yang dimiliki oleh kelas ini lebih besar dibandingkan dengan nilai false negative atau pun true negative. Sehingga baik LSTM maupun CNN mampu mengklasifikasi kelas ini dengan baik.

Pada kelas Feature Request, tidak ada perbedaan yang signifikan antara penggunaan metode CNN atau LSTM. Selain itu, data 300 menunjukan bahwa LSTM mampu

menunjukan performa akurasi terbaik dibandingkan dengan CNN serta data lainnya yang diuji coba, yaitu sebesar 98,78%. Namun, 884 Data menunjukan bahwa dengan bertambahnya data, nilai akurasi dari klasifikasi sedikit berkurang. Hal ini disebabkan oleh jumlah true negative dari kedua metode ini cenderung meningkat. Jika diamati, skor precision dan recall dari kelas ini merupakan salah satu skor kelas terburuk meskipun nilai akurasi yang diperoleh sangat baik. Nilai akurasi yang baik tersebut didukung oleh nilai true negative yang baik, tetapi tidak untuk true positive. Seluruh data uji dan kedua metode menunjukan bahwa ketidakseimbangan data dari proses training model mampu mempengaruhi kemampuan komputer untuk memahami pola kalimat pada kelas feature request.

Pada kelas Non Informative, uji coba yang dilakukan pada 544 data menunjukan performa terbaik dengan menggunakan CNN dibandingkan uji coba menggunakan data lainnya. Sedangkan uji coba dengan menggunakan 884 menunjukan performa terburuk untuk ke empat metode, terutama metode LSTM. Salah satu faktor yang menungkinkan kelas ini memiliki performa yang paling rendah dibandingkan kelas lainnya adalah karena kecenderungan kelas non informative yang lebih mudah terdeteksi ke dalam kelas lainnya. Jika dibandingkan performa antara metode LSTM dan CNN, terlihat bahwa sepintas metode CNN mampu mengungguli metode LSTM. Hal tersebut disebabkan oleh kemampuan LSTM yang hanya mampu mendeteksi true positive dan true negative. Sedangkan, pada data 884 nilai dari true positive lebih kecil dibandingkan dengan true negative. Sehingga nilai akurasi yang dihasilkan pun kurang optimal.

5. KESIMPULAN

Setelah diimplementasikan, hasil penelitian menunjukan bahwa Klasifikasi dokumen dapat dengan baik dilakukan dengan menggunakan Long Short Term Memory (LSTM) pada kategori Bug Report dengan nilai 93.1% untuk akurasi, 100% untuk precision, dan 93.1% untuk recall.

DAFTAR PUSTAKA

M. Hu, B. Liu, and S. M. Street, “Mining and Summarizing Customer Reviews,” in *Proceedings of The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang, “AR-Miner_:Mining Informative Reviews for Developers from Mobile App Marketplace,” in *Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 767–778.

L. V. G. Carre and K. Winbladh, “Analysis of User Comments_: An Approach for Software Requirements Evolution,” in *35th International Conference on Software Engineering (ICSE) 2013*, 2013, pp. 582–591.

Liu, Y. et al., 2013. Computer-Aided Design Identifying helpful online reviews : A product designer ’ s perspective. *Computer-Aided Design*, 45(2), pp.180–194. Available at: http://dx.doi.org/10.1016/j.cad.2012.07.008.

Liu, Y., Lu, W.F. & Loh, H.T., 2007. KNOWLEDGE DISCOVERY AND MANAGEMENT FOR PRODUCT DESIGN THROUGH TEXT MINING – A CASE STUDY OF ONLINE INFORMATION. , (August), pp.1–12.

Hu, M. & Liu, B., 2004. Mining and Summarizing Customer Reviews. In *Proceeding KDD ’04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 168–177.

Liu, B. et al., Reviewon Observer : Analyzing and Comparing Reviewons on the Web.

Xianghua, F. et al., 2013. Knowledge-Based Systems Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, 37, pp.186–195. Available at: http://dx.doi.org/10.1016/j.knosys.2012.08.003.

Maalej, W., 2015. Bug Report , Feature

Request , or Simply Praise ? On Automatically Classifying App Reviews. , pp.116–125.

Basari, A.S.H. et al., 2013. Reviewon Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization. *Procedia Engineering*, 53, pp.453–462.

Hamdan, H., Bellot, P. & Bechet, F., 2015. Lsislif : CRF and Logistic Regression for Reviewon Target Extraction and Sentiment Polarity Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. pp. 753–758.

Zhai, Z. et al., 2011. Clustering Product Features for Reviewon Mining. In *Proceeding WSDM '11 Proceedings of the fourth ACM international conference*. pp. 347–354.

Brody, S. & Elhadad, N., 2010. An Unsupervised Aspect-Sentiment Model for Online Reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*. pp. 804–812.

Wang, P. et al., 2016. Semantic Expansion using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification. *Journal of Neurocomputing*, 174, pp.806–814.

Hochreiter, S. & Schmidhuber, J., 1997. Long Short Term Memory. *Neural Computation*, 9, pp.1735–1780.